Stereo Object Tracking with Fusion of Texture, Color and Disparity Information

Olga Zoidi, Nikos Nikolaidis, Anastasios Tefas, Ioannis Pitas
Department of Informatics
Aristotle University of Thessaloniki
Box 451, Thessaloniki 54124, GREECE
tel: +30 2310 996361
{nikolaid, tefas, pitas}@aiia.csd.auth.gr

Abstract

A novel method for visual object tracking in stereo videos is proposed, which fuses an appearance based representation of the object based on Local Steering Kernel features and 2D color-disparity histogram information. The algorithm employs Kalman filtering for object position prediction and a sampling technique for selecting the candidate object regions of interest in the left and right channels. Disparity information is exploited, for matching corresponding regions in the left and right video frames. As tracking evolves, any significant changes in object appearance due to scale, rotation, or deformation are identified and embodied in the object model. The object appearance changes are identified simultaneously in the left and right channel video frames, ensuring correct 3D representation of the resulting bounding box in a 3D display monitor. The proposed framework performs stereo object tracking and it is suitable for application in 3D movies, 3D TV content and 3D video content captured by consuming stereo cameras. Experimental results proved the effectiveness of the proposed method in tracking objects under geometrical transformations, zooming and partial occlusion, as well as in tracking slowly deforming articulated 3D objects in stereo video.

Keywords: Stereo object tracking, Color disparity histograms, Local steering kernels

1. Introduction

Visual object tracking is an active research topic in computer vision, due to its wide range of applications, that include visual odometry [1], [2], robotic vision [3], human-centered interfaces [4], and surveillance systems [5]. The task of visual object tracking is a challenging one, since it has to overcome a number of difficulties, such as changes in illumination conditions, partial or total occlusion, self-occlusions, presence of cluttered background, non-smooth or complex object movements, object deformations, and noise. Traditionally, visual object tracking is performed in monocular videos captured by a single camera. In such systems, first, an initial object description is produced from an available object, then the new position of the object is detected based on some decision-making function and, then, the object description is updated, in order to incorporate the appearance changes, which arise from geometrical object transformations or other changes in the object appearance. The object representation may be appearance-based [6], feature-based [7], contour-based [8], or a combination of the above [9]. The decision function incorporates techniques for motion estimation [10], position prediction [11] and/or sub-sampling methods for reducing the computational cost of search [12]. Another tracking approach, mainly used in surveillance systems, is the discrimination of the object from the background through background subtraction [13]. Reviews of the recent advances on visual object tracking can be found in [14] and [15].

As described above, most stereo video tracking systems are set in constrained environments and use fixed position stereo cameras with known calibration parameters. However, in the last few years, with the availability of low-cost stereo video cameras and 3D display monitors, the amount of available stereo video data, from 3D cinema and 3D television to home-made 3D videos, has grown and will continue to grow exponentially. The majority of the available stereo video data are captured in unconstrained environments, with no information about the calibration parameters of the stereo system. Therefore, the development of tracking algorithms which exploit stereo information without extensive knowledge of camera calibration information is required.

In this paper, we present an appearance-based tracking algorithm, which exploits

stereo information obtained from the disparity maps of the left and right channels acquired by an uncalibrated stereo camera. The proposed framework combines a representation for the object texture based on Local Steering Kernel (LSK) descriptors [16], color information and disparity information. LSKs are local texture descriptors, which fit a Gaussian function over a local image region around an image pixel by elongating and steering the function according to the direction and intensity of the image edges. LSKs were employed successfully in object detection, where they were proven to be robust in small rotation and scale changes, as well as changes due to small object deformations. This robustness makes them capable of coping with the small object appearance changes in successive video frames during tracking. The algorithm requires no prior knowledge about the object, apart from its position in the first stereo frame. As tracking evolves, significant changes in object appearance are detected and stored in the object appearance model. At each stereo frame, first order Kalman filtering is performed for object position prediction from a video frame to the next one. The candidate stereo object positions are selected around the predicted position through sampling, using their 2-dimensional color-disparity histogram similarity to the stereo object in the previous stereo frame. The tracking procedure in the left and right channel videos is restricted by the stereoscopic geometry, in order to ensure that the displacement of the object Regions Of Interest (ROIs) from the left to the right video frame is equal to the mean disparity value of the object ROI. Moreover, the decision on the rotation and scale (zoom factor) of the object, as well as the update of the object model, are performed in the left and right channel video frames simultaneously. This ensures the stereo consistency of the resulting left and right channel object ROIs. Experimental results show that the proposed stereo tracking scheme is successful in tracking rigid and non-rigid objects that are subject to geometrical transformations, zooming, changes in the view angle, or illumination and partial occlusion, without prior knowledge of the object model. The method requires only the knowledge of the disparity maps. The novelties of the proposed approach are:

- the use of 2-dimensional color-disparity histograms for discriminating the object from its background,

- the combined use of the LSK descriptors and disparity information for deciding on the stereo object position concurrently on the left and right video channels,

- the use of a subsampling framework for reducing the number of candidate object ROIs during search in the next video frame.

The proposed method was tested in a number of stereo sequences captured from a stereo camera under different scenarios: generic object tracking with or without camera motion, partial or total object occlusion, varying light conditions, smooth and complicated object movement, changes in the object scale and in plane rotation and object deformations. Contrary to the state of the art stereo trackers that are either inapplicable in stereo sequences captured in unconstrained environments with unknown stereo system parameters, or they were not proven from experiments in the respective papers that they handle object deformations, occlusion and/or continuous changes in movement direction, the proposed stereo tracker was successful in all tracking scenarios.

The proposed stereo tracker extends the monocular tracker in [17], in the following ways. It operates on two channels, instead of only one, it employs 2-D color-disparity histograms for object separation from the background instead of color histograms, it does not perform exhaustive search for the object position, it employs more object appearances in the object model and it exploits the disparity information for searching for object change in scale. An early version of the proposed stereo tracker is introduced in [18]. The method introduced in this paper extends the paper in [18] in the following ways. It introduces a novel framework for searching possible changes in the object image scale and rotation, it provides a more detailed description of the algorithm, with auxiliary figures that highlight the novelties of the proposed method, it contains extended experimental results in many more video sequences and comparison to more state of the art trackers and it examines the significance of the disparity map quality in tracking accuracy.

The paper is organized as follows. The related work in the field is presented in Section 2. Section 3.1 describes the object position prediction and the candidate stereo object ROIs subsampling for object search in the next video frame. Section 3.2 presents the fusion of color and disparity information in 2D color-disparity histograms for fur-

ther reduction of the candidate object positions. Section 3.3 presents the object texture description based on LSKs. Section 3.4 describes the algorithm for the new stereo object position extraction and the object model update. Section 4 presents the experimental evaluation of the proposed method. Finally, conclusions are drawn in Section 5.

2. Related Work

The developments in video acquisition technology in the past decade led to an increasing use of multiple view systems in place of the monocular ones. For example, surveillance systems appeared that consist of one [19] or multiple stereo cameras [20] or multiple single-view cameras. These systems exploit the additional information obtained by exploiting the stereo geometry, namely the disparity information. Stereo tracking may be performed either in one or both the left and right video channels, whether they are rectified or not. In [19], the feature points on video frames acquired by cameras positioned high above the ground are projected on the 2D ground plane. The projected features are then clustered according to their 2D location and their height. The algorithm introduced in [21] performs stereo person tracking by transforming the camera-view depth images into plan-view statistical images and by employing adaptive statistical templates. Plan-view images are generated by projecting the foreground into a 3D point cloud in the camera coordinate system and by dividing the space into vertical bins, which correspond to plan-view pixels. In [22] and [23], multiple person tracking using a calibrated stereo camera is performed after background subtraction, by projecting the 3D point cloud to the 2D plan-view map introduced in [21]. Color information is also exploited, in order to separate the different persons. Unlike common surveillance stereo systems, in which the cameras are positioned high above the ground, in [22] the stereo camera is positioned at an under-head position. Plan-view maps of height and occupancy statistics are used for tracking in [24], where the sparse object appearance model, based on binary Gabor filters, is fused with stereo depth information.

Another application of stereo tracking is in pedestrian [25], [26] and vehicle [27]

tracking from moving vehicles. In these approaches, pedestrian tracking is performed in the entire video frame. The tracking procedure is divided into two steps. In the first step, background - foreground separation is performed. This is achieved through clustering on dense disparity maps and/or color intensity information [25, 26, 28] or through the selection of areas in which the disparity values are over a determined threshold [29]. Trained classifiers are then employed on foreground objects in order to discriminate the pedestrians from other foreground objects, based on intensity and/or disparity based feature extraction.

In [30], tracking is performed in the left and right video channels, by incorporating the epipolar constraints in a sum-of-squared differences (SSD) minimization problem. Multiple person tracking is performed in [31], with the fusion of color, gradient and depth information in multiple particle filters. The influence of the depth information in the algorithm varies, according to the density of the estimated disparity information. In the absence of disparity information, the tracker behaves like a monocular tracker. The tracking results of multiple stereo cameras are combined in the framework introduced in [20]. The tracking in each stereo camera is performed independently, exploiting shape, appearance and depth information. Tracking results of the various cameras are combined in a mixture model estimation approach. The proposed tracking framework achieves real-time tracking of multiple persons. The above mentioned methods are designed for person tracking only and they operate on stereo videos captured from one or more static stereo cameras with known camera calibration parameters, therefore, they cannot be employed for generic object tracking in stereo videos captured in unconstrained environments with camera motion and unknown calibration parameters. In [32] person tracking in the single video plus corresponding disparity map configuration is performed, based on face detection, skin color segmentation and disparity segmentation. The algorithm is also designed for person tracking only and operates on one channel of the stereo video.

Moreover, stereo tracking algorithms exist that exploit disparity information for object motion estimation. In [33], motion estimation is performed with the disparity motion vector (DMV), i.e., the difference between disparity maps which correspond to two successive stereo frames. The algorithm is based on the observation that movement

exists in the areas where the DMV value is high. The algorithm performs real-time tracking and does not require a calibrated camera. However, the algorithm assumes that the background is static and that only the object of interest is moving. This inhibits the application of the proposed tracker in stereo videos captured from moving cameras. Disparity information was combined with optical flow in [34] in the prediction of the 3D object velocity in the left and right channels. A variational framework for estimating the motion of points in the world coordinate system is introduced in [35]. The method performs independent estimation of both the depth and the 3D motion vector.

Finally, there are stereo object tracking algorithms that fuse information obtained from color videos and/or data from other sensors, such as depth videos captured from sensors like Kinect [36, 37, 38, 39]. In these algorithms, the disparity computation step is omitted since depth data are already available. In [36] the fusion of color and depth information is achieved by exploiting the calibration parameters of the stereo system that consists of the range sensor camera and the color camera. In [37] object tracking is perceived as a particle swarm optimization problem that estimates the object model parameters, i.e., position, orientation and articulation. In [38] depth information is employed for estimating the object's 3D model and fusion of contour information, obtained from the color camera and depth is expoited for correcting the estimated 3D object model. In [39] object tracking is perceived as foreground-background segmentation based on contour information, obtained by the color camera and depth information, obtained from the Kinect sensor. A novel method which fuses information obtained by range images produced from sensors with color information is introduced in [40], where the good behavior of the Iterative Closest Point (ICP) algorithm [41, 42] and the normal flow gradient constraint in object translation and object rotation, respectively, were exploited.

3. Method Description

The proposed algorithm performs tracking of rigid and deformable objects in 3D videos. The only information utilized is the left and right luminance channels and the

corresponding horizontal disparity maps. In stereoscopy, disparity is the difference (in pixels) of a projected 3D point as seen from the left and right camera [43]. The computation of quality disparity maps is an open issue in computer stereo vision with rich bibliography [44] [45] [46] [47]. However, this problem is outside the scope of this paper. Several reliable disparity estimation algorithms exist that operate on the left and right luminance channels, by finding matches between regions of the left and right channels. The proposed algorithm assumes that the disparity maps have been correctly estimated by using one of these algorithms and exploits them in the tracking framework.

The algorithm begins with the initialization of the object ROIs in the first frame of the left and right video channels. This initialization can be performed in two ways:

1. Perform object localization in the left and right frames independently manually, or through object detection.
2. Perform object localization in the left (right) frame. Object localization in the right (left) frame is performed through the mean disparity value in the left (right) frame.

We assume that the projections of the object on the left and right video frames have equal size. Therefore, in all initialization methods, care should be taken so that the chosen object ROIs have equal dimensions (in pixels). The ROIs in the left and right video frame which correspond to the object projections captured from the left and right cameras at the same time instance will be called stereo ROI pairs. For each video channel, an object model is defined as a stack containing the object instance in the first frame and the last $n-1$ object instances (ROIs) where significant change in the object appearance is detected, either due to an affine transformation, change in the object view angle and/or object deformation. The object instance in the first frame remains in the object model throughout the duration of tracking, while the last $n-1$ object instances in the model are updated every time a significant change is the object appearance is detected.

The overall method can be divided into two steps. First, object tracking is performed in each left-right luminance frame - disparity map configurations separately

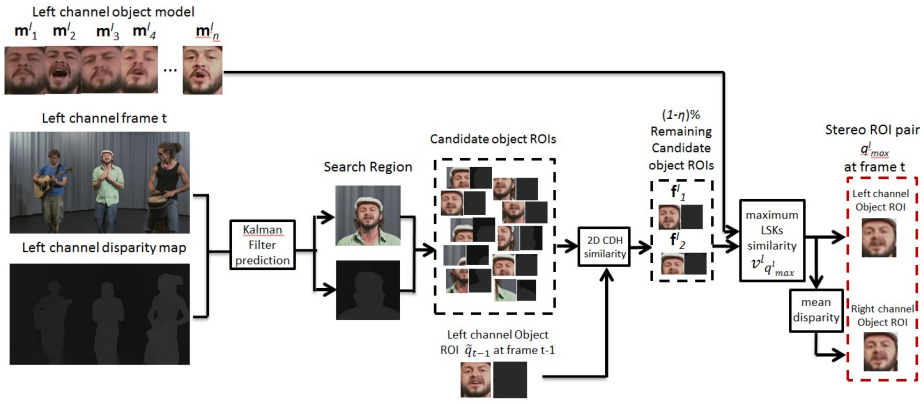


Figure 1: Extraction of candidate stereo object ROIs through search at the left channel.

and, then, the tracking results in the two left-right configurations are combined, in order to obtain the final decision of the object position. The block diagram of the first step is shown in Figure 1 for the left luminance frame - disparity map configuration. For each video frame - disparity map pair, the following iterative steps are performed:

- Prediction of the object position in the next video frame using Kalman filtering.
- Extraction of the search region, where the new object position will be searched in the next video frame and reduction of the candidate search positions through carefully chosen subsampling.
- Similarity computation through 2-dimensional video-disparity histograms of the reference object ROI with the selected candidate object ROIs for further reduction of the candidate object ROIs.
- Extraction of the texture descriptors (LSKs) of the candidate object ROIs.
- Decision on the final object ROI position in the frame based on the similarity of the candidate object ROIs with the object model templates (ROIs) for this channel.
- Selection of the object ROI in the corresponding frame of the other channel, through the mean disparity value of the object ROI in this channel and computation of its similarity to the object model of the other channel.

The application of the above procedure in the left and right video channels leads to the extraction of two candidate stereo object ROIs (stereo ROI pairs) on the two video frames, respectively. The stereo tracking algorithm then proceeds as follows:

- Computation of the maximum similarity of the candidate stereo ROI pairs in the current stereo video frame to the stereo ROI pairs in the object model based on LSK features and 2D color-disparity histograms.
- If the maximum similarity at the current frame is over a threshold, then the candidate stereo ROI pair with the maximum similarity is selected as the new stereo ROI pair. The threshold is determined as a percentage of the similarity of the detected stereo ROI pair in previous frame to the object model.
- If the maximum similarity is less than the threshold, then change in the object appearance, e.g. due to a geometrical transformation (scale or rotation), deformation, or change of the view angle is detected. In this case:
  - New candidate object ROIs around the stereo ROI pair's positions in the current left and right video frames are selected, for search of scaled and rotated versions of the object.
  - For the new candidate stereo ROI pairs, the similarities to the object model are computed.
  - The new candidate stereo ROI pair with the maximum similarity to the object model is selected as the new stereo ROI pair in the current left and right video frames.
  - The object model stacks in the left and right channels are updated by deleting the oldest stereo ROI pair and storing the new stereo ROI pair.

The following subsections present in detail the algorithm steps summarized above.

3.1. Search region extraction and subsampling

The algorithm commences with the initial prediction of the object position in the current frame of the left and right channel. Several methods exist for predicting the

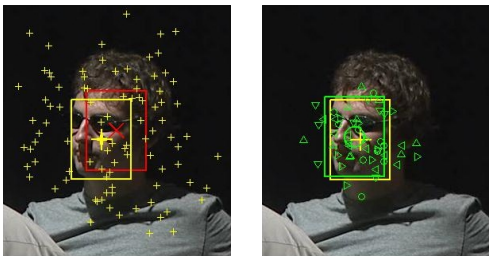
object motion state, such as, the Kalman filter, the extended Kalman filter, particle filters, the mean shift algorithm or by using optical flow. The prediction accuracy (and thus the selection of the algorithm that will be used for this purpose) does not play a significant role to the tracking performance, since it is employed only for determining the search region, in which the object will be searched. The first-order Kalman filter [48] was selected, due to its simplicity and reduced computational complexity and because, at it was proven experimentally, it is a good object motion predictor even for the case when the object direction changes constantly (second experiment in Section III). The object ROI position prediction is performed in each channel separately. The states $\mathbf{p}_t^l = [p_x^l, p_y^l, d_x^l, d_y^l]^T \in \Re^4$ and $\mathbf{p}_t^r = [p_x^r, p_y^r, d_x^r, d_y^r]^T \in \Re^4$ of the left and right video channel object ROIs at time t consist of the object ROI center (p_x, p_y) coordinates and the object ROI translation parameters (d_x, d_y), essentially measuring object ROI velocity in the previous video frame. A more detailed description of the Kalman filter model can be found in [17]. A different Kalman filter is used for predicting the new object ROI states $\hat{\mathbf{p}}_t^l$, $\hat{\mathbf{p}}_t^r$ on the left and right video channels, respectively.

The search region for the new object ROI in the next video frame is centered at the predicted position. The search region dimensions are equal to $R_x \times R_y = sQ_x \times sQ_y$, where $Q_x \times Q_y$ are the object ROI dimensions. s is a constant which determines the size of the search region, according to the object speed (fast/slow motion). Typical values of s are 1.5 (for slow smooth movements), 2.0 and 2.5 (for fast, complex movements).

After search region extraction, search region subsampling is performed in order to select n candidate object ROIs, instead of performing exhaustive search, in order to increase tracking speed. The set of n candidate object ROIs positions $\mathbf{Y}_t^l$, $\mathbf{Y}_t^r$ (Figure 2(a)) in the t-th left and right video frames, respectively, are selected randomly according to:

$$\mathbf{Y}_t^j = \{\mathbf{y}_t^{j1}, \ldots, \mathbf{y}_t^{jn}\} \sim N(\hat{\mathbf{p}}_t^j, \mathbf{\Sigma}), \qquad j = r, l, \tag{1}$$

where $\mathbf{\Sigma} = \text{diag}[R_x/m, R_y/m]$. We typically choose $m = 4$. In the left and right search regions, the best candidate object ROI states $\tilde{\mathbf{p}}_t^l$, $\tilde{\mathbf{p}}_t^r$, corresponding to left/right object ROIs having maximum similarity to the object model instances, are extracted according to the procedure described in the subsequent sections 3.2 to 3.4. Then, a more refined search around the best candidate object ROIs position is performed by extract-



(a) (b)

Figure 2: Search region subsambling: (a) Selection of n candidate object ROIs positions (crosses) centered at the predicted object position (X). The most probable candidate object ROI is denoted with the bounding box centered at the large cross. (b) Extraction of $\rho = 5$ sets of candidate object ROIs positions centered at the most probable candidate object (large cross) for search of the object change in scale and rotation. The final object position is denoted with the bounding box centered at the large circle.

ing ρ sets of n' secondary candidate object ROIs positions (Figure 2(b)), according to:

$$\mathbf{Y'}_t^j = \{\mathbf{y'}_t^{j1}, \ldots, \mathbf{y'}_t^{jn'}\} \sim N(\tilde{\mathbf{p}}_t^j, \mathbf{\Sigma}'), \qquad j = r, l, \tag{2}$$

where $\mathbf{\Sigma}' = \text{diag}[R_x/m', R_y/m']$, $m' > m$. We typically choose $m' = 10$. A subset of the ρ secondary candidate object ROIs is used for detecting possible object image zoom-in and zoom-out in various scales, another subset for clockwise and another for counterclockwise in-plane rotation in various degrees, and the final for more accurate object ROI position detection.

In each channel a different Kalman filter and subsampling was employed, resulting in different candidate object positions. The reason for this that we perform a more detailed object search without increase in the algorithm computational complexity, with respect to searching for the new object position in corresponding (through disparity) candidate object positions in the left and right channel. For each channel, the described subsampling technique reduces the computational complexity of the algorithm with respect to exhaustive object search from $O\left(\rho(R_x - Q_x + 1)(R_y - Q_y + 1)c\right)$ to $O((n + \rho n')c)$, where c is the computational cost for computing the candidate object similarity to the object model. For example, for an object having ROI size 30×30 pixels and $s = 1.5$, by setting $n = 100$, $n' = 10$ and $\rho = 5$, the computational cost in each left/right video channel is reduced from $O(1280c)$ to $O(150c)$, i.e., by one order

of magnitude.

3.2. Color-disparity similarity

In stereoscopic systems, disparity is the most essential additionally available information, compared to monocular systems. The disparity values provide an intuitive notion of the relative object 'depth' from the camera (to a scale factor, when the camera axis are parallel). More precisely, the larger the disparity value, the closer the object position is to the camera. Therefore, the majority of the state-of-the-art algorithms, which perform object tracking in stereo videos exploit disparity information, in order to find the new object position. A common way of exploiting disparity information for distinguishing the tracked object from the background is by performing disparity segmentation on the disparity map. The disadvantage of disparity segmentation is that it does not take into account color information, therefore it cannot easily discriminate between objects that lie in the same distance from the camera. Another way for discriminating the object from the background is by performing disparity-histogram similarity check followed by color-histogram similarity check [32]. This way, both disparity and color information are employed, however the spatial correlation between the disparity and color information is not exploited. In the proposed method, the color and disparity correlation is fully exploited by their combination in a 2-dimensional color-disparity histogram, as shown in Figure 3. Generally, the object color and disparity histograms are not constant throughout the video duration but vary, due to illumination variations, changes in the view point and/or object movement towards or away from the camera. However, between two consecutive frames, this change can be considered to be rather small. Therefore, we can reduce the number of the candidate object ROIs at frame t by discarding the ones with the lowest 2-D color-disparity histogram similarity to the detected object ROI at frame $t-1$.

Each candidate object ROI is split into its three RGB color channels and for each channel the 2-D color-disparity histogram $\mathbf{H}_R, \mathbf{H}_G, \mathbf{H}_B \in \Re^{n_c \times n_d}$ is computed. As a result, three 2-D histograms correspond to each object ROI. The 2-D histograms are constructed by selecting n_c bins for the color and n_d bins for the disparity information. The color bins widths are selected uniformly, while the disparity bins are selected

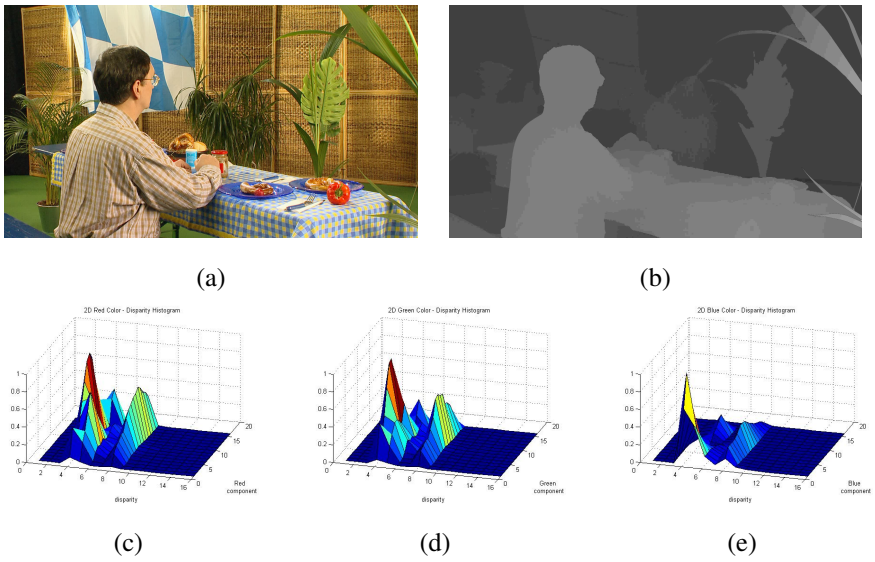


Figure 3: a) Luminance channel, b) corresponding disparity map, c) 2-D Red channel-disparity histogram, d) 2-D Green channel-disparity histogram, e) 2-D Blue channel-disparity histogram.

according to the following procedure:

- Find the minimal and maximum disparity value of the first frame.
- Set the width of the first bin from 0 to the minimal value.
- Set the width of the n_d-th bin from the maximum value in the first frame to the maximum disparity value in the whole video.
- Set the width of the remaining bins uniformly in the range from the minimal to the maximum disparity value.

The 2-D histograms for the R, G and B component are depicted in Figure 3, for $n_c = n_d = 16$.

The 2-D color-disparity histograms were compared by computing their cosine similarity as follows. The 2-D color-disparity histograms $\mathbf{H}_R, \mathbf{H}_G, \mathbf{H}_B$ are column-stacked into 1-D color-disparity histograms $\mathbf{h}_R, \mathbf{h}_G, \mathbf{h}_B \in \Re^{n_c \cdot n_d}$ and they are compared to the corresponding object histograms $\hat{\mathbf{h}}_R, \hat{\mathbf{h}}_G, \hat{\mathbf{h}}_B \in \Re^{n_c \cdot n_d}$ of the previous frame via

the cosine similarity:

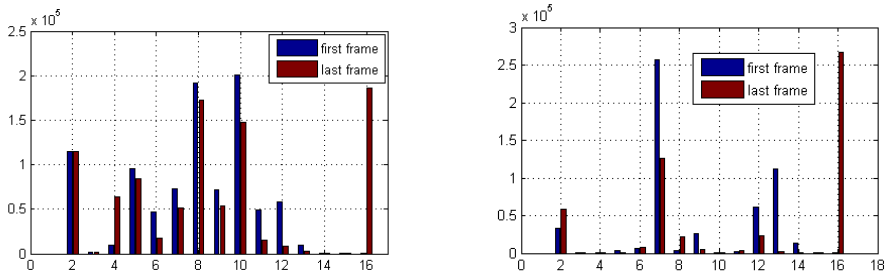
$$c_k(\mathbf{h}_k, \hat{\mathbf{h}}_k) = cos(\theta) = \frac{< \mathbf{h}_k, \hat{\mathbf{h}}_k >}{\|\mathbf{h}_k\| \|\hat{\mathbf{h}}_k\|} \in [-1, 1], \qquad k = R, G, B, \tag{3}$$

where $< \cdot >$ is the inner product, $\| \cdot \|$ denotes the Euclidean norm and θ is the angle of the two vectors. Cosine similarity between two vectors is an indicator of wether the vectors point in the same direction and is computed by calculating the cosine of the angle the two vectors form. The total histogram similarity is computed as:

$$S = \sum_{k=R,G,B} \frac{c_k^2}{1 - c_k^2} \in [0, +\infty). \tag{4}$$

The total histogram similarity (4) is computed for every candidate object ROI and the $\eta\%$ candidate objects with the lowest histogram similarity are discarded. The threshold η may vary according to the shape of the 2-D histogram, or it may be fixed. At the conducted experiments, we considered a fixed value $\eta = 80\%$, meaning that only 20% of the selected candidate object ROIs will be further examined for being the new object ROI.

This proposed selection of the disparity bins performs good foreground-background separation, when the video sequence disparity does not change much. In the case when the object moves too close to the cameras, all of its disparity values will lie in the last bin. This will still discriminate the object from the background, whose disparity values will lie in the other bins too. Therefore, we still expect the algorithm to perform well. During camera zoom, we expect the background and the object disparity histograms to be similar because the disparity range of the whole scene changes and thus disparity alone will not contribute to the foreground-background separation. However, in such a case, most of the frame disparity values will lie in the first or the last histogram bin. In this case, the object discrimination from the background will still be feasible from the color component in the 2D-CDH. This highlights the ability of 2D-CDHs to discriminate the object from the background, when either the color or the disparity histogram similarity fails to do so. Examples of disparity histogram changes due to object motion towards the camera or extreme camera zoom-in are depicted in Figure 4.



(a) (b)

Figure 4: Change in disparity histograms in the case of a) object motion towards the camera and b) extreme camera zoom-in.

3.3. Use of object texture in tracking

During tracking, we assume that the object appearance does not change significantly from frame t to frame $t+1$. Therefore, various image texture descriptors can be employed for finding which of the remaining candidate object ROIs is most similar to the object appearance model. In this paper, we have chosen to use Local Steering Kernel (LSK) descriptors, which were employed successfully in [16] for generic object detection and in [17] for monocular object tracking, due to their robustness in the appearance changes that the object undergoes between successive frames. LSKs are local image texture descriptors which measure the similarity of an image pixel to its surrounding ones. LSKs fit a Gaussian kernel function over a local area of size $P \times P$ pixels around the central pixel by elongating and steering the Gaussian kernel, according to the local image texture morphology, by taking into account the pixel luminance value difference and the spatial distance between the center pixel and its neighbors:

$$k(\mathbf{p}_l - \mathbf{p}) = \frac{\sqrt{\det(\mathbf{C}_l)}}{2\pi} \cdot \exp\left\{-\frac{(\mathbf{p}_l - \mathbf{p})^T \mathbf{C}_l (\mathbf{p}_l - \mathbf{p})}{2}\right\}, \qquad l = 1, \ldots, P^2, \quad (5)$$

where $\mathbf{p}, \mathbf{p}_l \in \mathcal{Z}^{+2}$ are the vectors of the center pixel and the neighboring pixel coordinates, respectively. The shape of the transformed Gaussian function is determined by the covariance matrices $\mathbf{C}_l$ of the image gradients $\mathbf{J}_l$ defined in local areas of size $P \times P$ centered at the neighboring pixels $\mathbf{p}_l$:

$$\mathbf{J}_l = \left[\mathbf{z}(\mathbf{p}_1^l), \mathbf{z}(\mathbf{p}_2^l), \ldots, \mathbf{z}(\mathbf{p}_{P^2}^l)\right]^T, \quad (6)$$

where $\mathbf{z}(\mathbf{p}_i^l) = [z_x(\mathbf{p}_i^l), z_y(\mathbf{p}_i^l)]^T$, $i = 1, \ldots, P^2$ denote the image gradient vectors along x and y axes at the i^{th} neighboring pixel of $\mathbf{p}_l$.

For a given pixel $\mathbf{p}$, equation (5) is computed P^2 times, one for each pixel in a neighborhood of size $P \times P$, resulting in a vector representation $\mathbf{k}(\mathbf{p}) \in \Re^{P^2 \times 1}$ of the local image texture. The LSK vectors become invariant in illumination variations by employing L_1 normalization:

$$\mathbf{k}_N(\mathbf{p}) = \frac{\mathbf{k}(\mathbf{p})}{|\mathbf{k}(\mathbf{p})|_1}, \tag{7}$$

where $|\cdot|_1$ is the L_1 norm. By ordering the normalized vectors column-wise, we produce the LSK feature matrix of the object ROI $\mathbf{K} \in \Re^{P^2 \times N_x N_y}$, where N_x, N_y are the object ROI dimensions. The above mentioned LSK feature matrices are computed for the candidate object ROIs with the higher 2-D color histogram similarity as described in subsection 3.2. Finally, dimensionality reduction through PCA is employed, so that only the most significant information is retained. PCA is employed on the object LSK feature matrix at the first video frame. Then, the candidate object ROIs feature matrices are projected onto the space generated by the projection matrix derived from the first video frame, creating the salient LSK feature matrices $\mathbf{F} \in \Re^{d \times N_x N_y}$, where d is the reduced dimension of the LSK feature vectors. In our experiments, we set $d = 3$.

3.4. Object localization and model update

The procedure described in subsections 3.1, 3.2, i.e., the selection of the candidate object ROIs, is performed for the left and right channels, independently. During this procedure, the stereo information employed is the disparity map, fused with the RGB color information into 2-dimensional color-disparity histograms. However, the relation between the left and right channel video frames is not fully exploited yet. The geometry of the stereoscopic camera system implies that, apart from the case where the object is not visible from one of the cameras of the stereo system, any changes in object appearance due to geometrical transformations, (zooming, rotation), change of view angle and/or object deformations occur simultaneously to both the left and right video channels. The proposed tracking framework fully exploits this stereo information by coupling the tracking results in the left and right videos as described below.

First, the salient LSK feature matrices $\mathbf{F}_q^l, \mathbf{F}_q^r \in \Re^{d \times Q_x Q_y}$ of the candidate object ROIs in the left and right videos, where q denotes the index of the candidate object ROIs, are extracted and their similarity to the left and right video object model instances

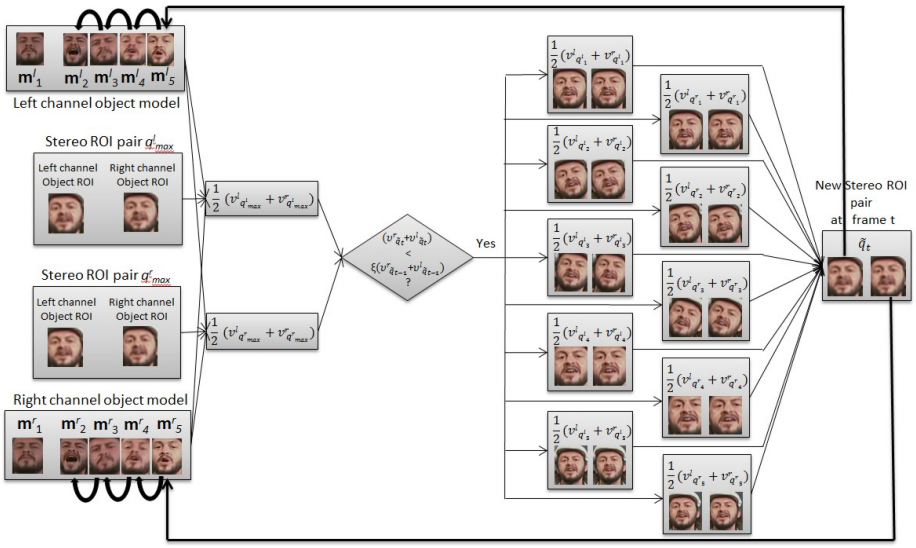


Figure 5: Update of the object models in the left and right channels.

are computed. As tracking evolves, the object model is updated by inserting the latest object instance in the stack and deleting the $(n-1)-th$ oldest object instance from it, as shown in Figure 5. The object instance in the first frame remains in the object model stack throughout the tracking procedure. This makes the algorithm able to keep tracking the object , when its original view angle reappears, after significant and long-term changes in the object appearance.

We define as $\mathbf{M}_k^l, \mathbf{M}_k^r, \in \Re^{d \times Q_x Q_y}$, $k = 1, \ldots, n$ the salient LSK feature matrices of the left and right object model instances, respectively. The object model instances are initialized with the object instance in the first left ($\mathbf{M}_1^l$) and right ($\mathbf{M}_1^r$) frame, i.e., $\mathbf{M}_1^l = \mathbf{M}_2^l = \cdots = \mathbf{M}_n^l$ and $\mathbf{M}_1^r = \mathbf{M}_2^r = \cdots = \mathbf{M}_n^r$. The resemblance of the $q-th$ candidate object ROI to the $k-th$ object model instance is estimated using cosine similarity:

$$c_{qk}^j(\mathbf{f}_q^j, \mathbf{m}_k^j) = cos(\theta) = \frac{< \mathbf{f}_q^j, \mathbf{m}_k^j >}{\|\mathbf{f}_q^j\| \|\mathbf{m}_k^j\|} \in [-1, 1], \qquad k = 1, \ldots, n, \qquad j = l, r, \qquad (8)$$

where $\mathbf{f}, \mathbf{m} \in \Re^{dQ_x Q_y}$ are the column-stacked vectors of the salient LSK feature matrices. The overall resemblance of the $q-th$ candidate object ROI to the object

model is defined as the weighted sum:

$$v_q^j = \lambda \frac{c_{q1}^{j2}}{1 - c_{q1}^{j2}} + \frac{1-\lambda}{n} \sum_{k=2}^{n} \frac{c_{qk}^{j2}}{1 - c_{qk}^{j2}} \in [0, +\infty), \qquad j = l, r, \tag{9}$$

where $0 \leq \lambda \leq 1$ is a weight parameter on the similarities of the candidate object ROIs to the object model instance in the first frame. A typical value for λ is 0.5. For the left (right) video frame, the overall resemblance (9) of the candidate object ROIs to the left (right) object model instances are computed and stored in the vector $\mathbf{v}^l = [v_1^l, \ldots, v_n^l]^T \in \Re^n$ ($\mathbf{v}^r \in \Re^n$), where n is the number of candidate object ROIs. The candidate object ROI $q_{max}^l = \arg\max_q\{\mathbf{v}^l\}$ ($q_{max}^r = \arg\max_q\{\mathbf{v}^r\}$), where $\arg\max_q\{\cdot\}$ denotes the index of the vector entry with the maximum value, with the largest resemblance to the object model instances is selected and, through its mean disparity value, the corresponding object ROI in the right (left) channel is detected and its similarity $v_{q_{max}^l}^r$ ($v_{q_{max}^r}^l$) to the object model instances in the right (left) channel is extracted. We finally have two likely stereo object ROIs: the first resulting from the object search in the left channel and the other resulting from the object search in the right channel. The stereo ROI pair with the maximum average similarity to the stereo ROI pairs in the left/right object models is selected as a probable stereo ROI instance:

$$\tilde{q} = \arg\max_{q_{max}^k} \left\{ \frac{1}{2} \left(v_{q_{max}^k}^r + v_{q_{max}^k}^l \right) \right\}, \quad k = l, r, \tag{10}$$

If the object ROI q_{max}^l (q_{max}^r) lies in the left (right) border of the left (right) channel and is visible only in the left (right) channel, then there doesn't exist a corresponding object ROI in the right (left) channel and, thus, we take into account only the similarity to the ROI in the left (right) object model by setting $v_{q_{max}^l}^r = v_{q_{max}^l}^l$ ($v_{q_{max}^r}^l = v_{q_{max}^r}^r$).

The similarity of the probable stereo ROI pair $\tilde{q}_t$ at frame t to the object model instances is compared to the similarity of the stereo ROI pair $\tilde{q}_{t-1}$ at frame $t-1$ to the object model instances. If $\tilde{q}_t < \xi \tilde{q}_{t-1}$, where ξ is a predetermined threshold, then a change in the object appearance is detected. In the conducted experiments, we set $\xi = 0.9$, meaning that a change in the object appearance is detected when the stereo ROI pair similarity to the object model instances between two consecutive video frames drops under 90%. When a change in the object appearance is detected, a more detailed search around the position of the stereo ROI pair $\tilde{q}_t$ is performed. First, n' candidate object ROIs at the left and right video frames are selected according to (2).

Then, for each candidate object ROI, the color-disparity similarities (4) and overall similarities (9) to the object model instances are computed, according to the procedure described in subsections 3.2 and 3.3 and equations (8)-(9). The similarities of the candidate object ROIs to the object model instances for the left and right channel are stored in vectors $\mathbf{v}_1^l \in \Re^{n'}$ and $\mathbf{v}_1^r \in \Re^{n'}$, respectively. The in-plane rotation of the object by $\pm\varphi$ degrees is examined by rotating the left and right frame $\mp\varphi$ degrees around the predicted object ROI position and by extracting n' candidate object ROIs according to (2), producing the similarity vectors $\mathbf{v}_2^l, \mathbf{v}_2^r \in \Re^{n'}$ (for $+\varphi$ degrees) and $\mathbf{v}_3^l, \mathbf{v}_3^r \in \Re^{n'}$ (for $-\varphi$ degrees). Finally, the change in the object scale by $\pm s\%$ due to camera zooming or object movement towards and away from the camera is examined by resizing the left search region by $\mp s\%$ and by selecting n' candidate object ROIs according to (2). The similarities of the candidate object ROIs to the object model instances are stored in vectors $\mathbf{v}_4^l, \mathbf{v}_4^r \in \Re^{n'}$ (for $s\%$ zoom-in) and $\mathbf{v}_5^l, \mathbf{v}_5^r \in \Re^{n'}$ (for $s\%$ zoom-out). Typical values for φ and s are 10 degrees and 10%, respectively.

Given $q_\kappa^j = \arg\max_q\{\mathbf{v}_\kappa^j\}$, $\kappa = 1\ldots, 5$, $j = r, l$, the new position of the object ROI in the left and right channel is computed by:

$$\tilde{q} = q_{\kappa'}^{j'} = \arg\max_{q_\kappa^j}\left\{\frac{1}{2}\left(v_{q_\kappa^j}^r + v_{q_\kappa^j}^l\right)\right\}, \qquad \kappa = 1, \ldots, 5, j = r, l, \tag{11}$$

provided that the condition:

$$\text{mean}(\mathbf{v}_{\kappa'}^{j'}) > \max\{\text{mean}(\mathbf{v}_1^l), \text{mean}(\mathbf{v}_1^r)\} \tag{12}$$

is satisfied, where $\text{mean}(\mathbf{v})$ denotes the mean value of the elements of vector $\mathbf{v}$. Moreover, for the case of object zoom in, the mean disparity value of the object pair should be increased by $s\%$ and in the case of object zoom out, the object pair mean disparity value must be decreased by $s\%$. If these conditions are not satisfied, then

$$\tilde{q} = \arg\max_{q_1^k}\left\{\frac{1}{2}\left(v_{q_1^k}^r + v_{q_1^k}^l\right)\right\}, \quad k = r, l. \tag{13}$$

The object models for the left and right videos are then updated with the corresponding object ROIs of the new stereo object. In the next stereo frame, the stereo object will be searched at the scale and angle of the last stored stereo object.

4. Experimental results

4.1. Experimental setup

The performance of the proposed tracking scheme was tested in nine videos captured by a stereo camera[1]. The video resolution was 1920×1080 pixels per channel. The employed method for extracting the disparity maps in videos 1-6 is described in [49], [50], while the disparity maps of videos 7-9, which were captured with a consumer-grade stereo camera, were extracted using the method described in [51]. The initialization of the tracking algorithm was accomplished with the object detector described in [16]. The search region size is $R_x \times R_y = 1.5Q_x \times 1.5Q_y$, where $Q_x \times Q_y$ are the downscaled object dimensions, which are selected for each experiment, as shown in Table 4.1. The prediction model and parameters of the Kalman filter are the ones given in [17]. The number of random candidate object ROIs positions in the left and right frame is $n = 100$. Their position has a 2D normal distribution, centered at the object ROI prediction with covariance matrix diag$[R_x/4, R_y/4]$. A more refined search for object scale by $S\% = \pm 10\%$ and rotation by $\phi = \pm 10$ degrees is performed when the similarity to the object model between two consecutive images drops under $\xi = 90\%$. Then, $\rho = 5$ sets of $n' = 10$ random candidate object ROI predictions are selected having a normal distribution centered at the best candidate object ROI pair with covariance matrix diag$[R_x/10, R_y/10]$. The 2-D color-disparity histograms have $n_c \times n_d = 16 \times 16$ bins. $\eta = 80\%$ of the candidate object ROI pairs are discarded through 2D-CDH similarity. For the remaining candidate object ROI pairs the LSK features are computed for window size $P \times P = 3 \times 3$ and reduced dimensionality $d = 3$. Finally, the object model consists of $k = 5$ object instances, i.e., the initial object instance, with weight $\lambda = 0.5$, plus four additional object instances.

4.2. Qualitative evaluation

In the first stereo sequence (Figure 6) the tracking performance was tested in a simple tracking scenario of a face, having small scale variations and slow view angle

[1]The tracking results are available at `https://www.dropbox.com/sh/tgbz7izcj0mkd40/WY1RkY6BUe`

Table 1: down-scaled object dimensions for the videos 1-9.

video	Q_x	Q_y	video	Q_x	Q_y
video 1	42	47	video 6	27	33
video 2	31	24	video 7	34	40
video 3	58	55	video 8	19	77
video 4	30	32	video 9	29	34
video 5	25	18			

changes. The second stereo video (Figure 7) depicts an object (a face) that performs fast and complex movements, with frequent changes in motion direction and view angle. In the third experiment, the tracking performance was tested on tracking a face, with small changes in orientation under severe occlusion (Figure 8). The problem of tracking an object (a face), subject to partial occlusion and illumination variations, caused by the shadows covering parts of the object, was examined in the fourth stereo sequence (Figure 9). In the fifth stereo video (Figure 10), tracking performance was tested when tracking non-rigid objects, i.e., a human hand performing gestures. The sixth stereo video (Figure 11) shows an object (a face) with fast movement and partial occlusion. The task of tracking an object (the woman's bag) that has similar color and disparity to the background (the woman's coat) and performs smooth movement was examined in the seventh stereo sequence (Figure 12). In the eighth video (Figure 13), the task was to track a person's body that is totally occluded by another person over a number of video frames. Finally, in the ninth stereo sequence (Figure 14), the objective was to track a rigid object (a helmet) that performs smooth movement with continuous changes in the view angle. The tracking results in Figures 6-14 show that the proposed stereo tracking algorithm is robust in scale, in-plane and view angle variations of rigid and non-rigid objects having small deformations, under partial or total occlusion. In general, the algorithm handles occlusions that occur gradually, by incorporating (for a small temporal interval) the partial object occlusion in the object model. When the occlusion is over, the increased weight of the object instance in the first frame forces the tracker to follow the object of interest.

Figure 6: Tracking results on a rigid object with small scale changes and view angle variations.

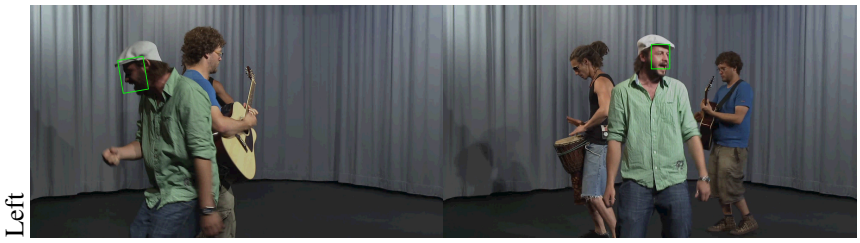


Figure 7: Tracking results on a fast moving object with constant changes in direction.

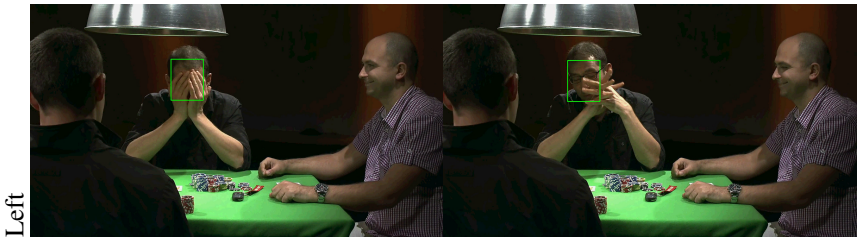


Figure 8: Tracking results on an occluded object.

4.3. Quantitative evaluation

The proposed framework takes into consideration the information obtained from both the right and left videos, leading to a disparity-consistent representation of the tracking result, i.e., the bounding boxes in the left and right frames are well linked by the respectively disparity values. The significance of the incorporation of disparity information in the stereo tracking algorithm is examined by comparing the performance of the stereo LSK tracker to the monocular LSK tracker [17]. The monocular LSK

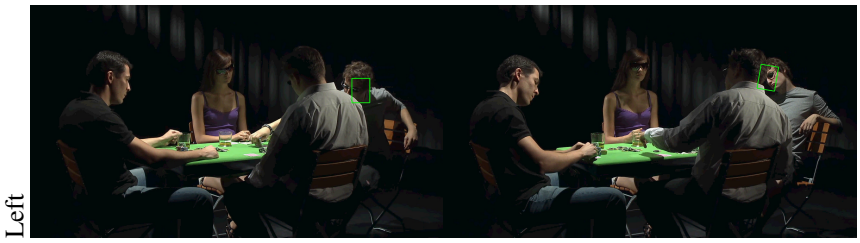


Figure 9: Tracking results on a occluded object with lighting variations.

Figure 10: Tracking results on an articulated object.

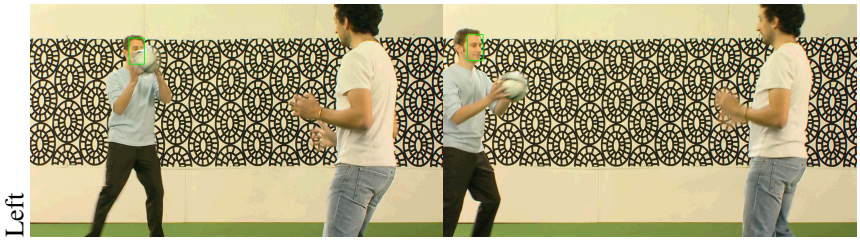


Figure 11: Tracking results on a fast moving object with partial occlusion.

tracker operates on the left and right video channels independently, based on color-histogram information (instead of the 2-D color-disparity histogram of the stereo case) and LSK descriptors. Moreover, the stereo tracking performance will be compared to the performance of four state-of-the-art monocular appearance-based trackers, namely CH tracker [52] that is based on color histogram information and particle filtering, L1 tracker [53] that is based on sparse representation of the object appearance and particle filtering, MIL tracker [54] that is based on online multiple instance learning of

Figure 12: Tracking results on a rigid object with similar color and disparity to a background object.

Figure 13: Tracking results on a rigid object under total occlusion.

Figure 14: Tracking results on a rigid object with continuous changes in view angle.

an adaptive appearance model for the object and finally CT tracker [55] that performs real-time compressive tracking.

The stereo and monocular trackers were tested on videos 1-9. The length of each video in frames is shown in the second column of Table 2. A quantitative evaluation of the tracking frameworks is accomplished by measuring the *Stereo Frame Detection Accuracy* (SFDA), which is the average overlap area between the tracked object in the left and right frame and the corresponding ground truth. It is an extension of the Frame

Detection Accuracy (FDA) measure proposed in [56]. Given the tracked object regions T^l, T^r in the left and right frame of the stereo video and G^l, G^r the corresponding ground regions, the SFDA at stereo frame t is defined as:

$$SFDA(t) = \frac{1}{2N_t^l} \sum_{i=1}^{N_t^l} \frac{|G_i^l(t) \cap T_i^l(t)|}{|G_i^l(t) \cup T_i^l(t)|} + \frac{1}{2N_t^r} \sum_{j=1}^{N_t^r} \frac{|G_j^r(t) \cap T_j^r(t)|}{|G_j^r(t) \cup T_j^r(t)|}, \tag{14}$$

where $|G|$ denotes the area of region G and N_t^l, N_t^r denote the number of objects in the left and right frame, respectively. The *Average Tracking Accuracy* (ATA) is defined as the average SFDA evaluated over all the stereo frames in a video:

$$ATA = \frac{1}{N} \sum_{t=1}^{N} SFDA(t), \tag{15}$$

where N is the number of stereo frames in the video. Moreover, the Overall Tracking Accuracy (OTA) is defined as:

$$OTA = \frac{1}{N_T} \sum_{i=1}^{k} N_i ATA_i, \tag{16}$$

where k is the total number of videos (in our case $k = 9$), N_i is the number of stereo frames of the $i - th$ video and $N_T = \sum_{i=1}^{k} N_i$ denotes the total number of stereo frames.

Figure 15 depicts the SFDA of the stereo and the monocular trackers versus time for the videos of the case studies 1-9, while the ATA and the OTA of the stereo and the monocular trackers for the nine videos of the experiments are shown in Table 2. We notice that, in seven out of nine videos, the stereo LSK tracker achieves a better average tracking accuracy than the monocular trackers. In video 4, the ATAs of the monocular MIL and LSK trackers are 11% and 3% better than the ATA of the proposed stereo LSK tracker, respectively, and in video 5, the ATA of the monocular L1 tracker is 6% better than the ATA of the stereo LSK tracker. By examining the ATA values of all trackers in all videos we notice that, the ATA of the stereo tracker is more consistent than the ATA of the monocular trackers that perform very well only in few videos and completely fail in other videos. This is manifested by the fact that the OTA of the proposed stereo LSK tracker is 14%, 41%, 42%, 37% and 28% better than the OTA of the monocular LSK, CH, L1, CT and MIL trackers, respectively and by the fact that the proposed stereo tracker has the smallest ATA variance (0.0029), as shown in the

Table 2: ATA and OTA of the trackers for the videos in case studies 1-9.

	length	stereo LSK	monocular LSK	CH	L1	CT	MIL
video 1	930	**0.6324**	0.6069	0.2882	0.5580	0.3646	0.5284
video 2	629	**0.5633**	0.5313	0.0555	0.0994	0.4320	0.1077
video 3	689	**0.7136**	0.4671	0.4877	0.3422	0.3064	0.5776
video 4	500	0.6737	0.6962	0.6120	0.5901	0.5610	**0.7537**
video 5	500	0.6574	0.5498	0.4754	**0.6975**	0.5481	0.3266
video 6	500	**0.6808**	0.5236	0.4940	0.3653	0.1380	0.4322
video 7	165	0.7554	**0.7558**	0.6440	0.1568	0.5386	0.3881
video 8	95	**0.5187**	0.2881	0.1938	0.0542	0.1835	0.1983
video 9	545	**0.5993**	0.4972	0.3318	0.0242	0.5528	0.5611
OTA		**0.6459**	0.5553	0.3811	0.3707	0.4070	0.4617
variance		**0.0029**	0.0074	0.0299	0.0526	0.0189	0.0345

last row of Table I. Therefore, the proposed stereo tracker is more robust when used in a wide variety of videos.

4.4. Significance of 2D color-disparity histograms

The significance of the proposed 2-D color-disparity histograms is examined by comparing the tracking performance of the proposed stereo tracker when 1-D color histograms (stereo CH LSK tracker) are used. The 1-D color histograms have 256 bins, like the ones employed in [17], also because the employed 2D-CDH contains 256 bins, (16 for color $\times$ 16 for disparity). The results are shown in Table 3. It is shown that, in 8 out of 9 cases the use of 2-D color-disparity histograms leads to more robust tracking accuracy. In case study 8 the use of color histogram achieves better tracking accuracy, since the object being tracked (the man's body) is more colorful than the objects in the other videos and requires more than 16 bins for the color component. In such case, 2-D color-disparity histograms with more than 16 bins for the color component can be used.

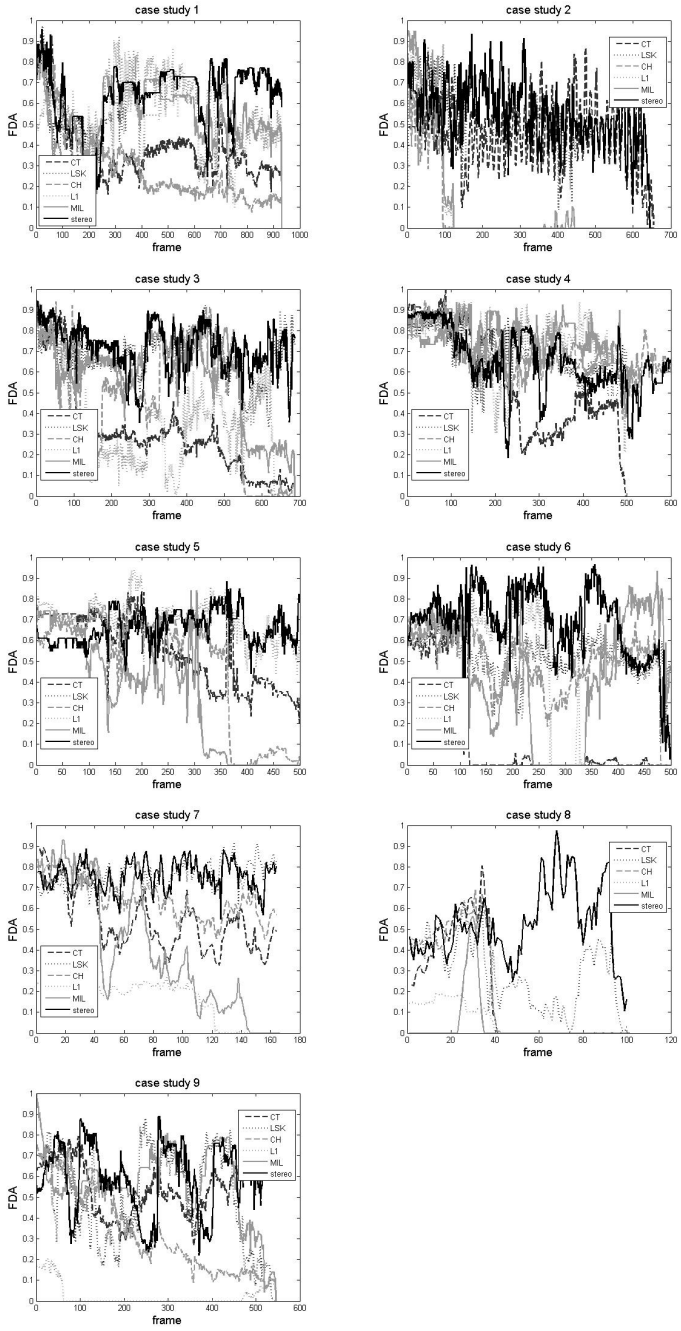


Figure 15: FDA of the stereo and monocular trackers for the videos in case studies 1-9.

Table 3: ATA of the stereo tracker when 2-D color-disparity histograms (2D CDH) and 1-D color histograms (1D CH) are employed.

ATA	video 1	video 2	video 3	video 4	video 5	video 6	video 7
2D CDH	0.6324	0.5633	0.7136	0.6737	0.6574	0.6808	0.7554
1D CH	0.5711	0.5423	0.6403	0.6043	0.5866	0.5408	0.7168

ATA	video 8	video 9
2D CDH	0.5187	0.5993
1D CH	0.5395	0.5418

4.5. Significance of disparity maps quality

Finally, we test the performance of the proposed stereo tracker in cases where the available disparity map contains errors. This situation occurs, e.g., when the disparity map is extracted using rather poor disparity estimators, e.g., the fast disparity estimator [57]. Example disparity maps estimated using a high accuracy and a lower accuracy disparity estimation method are presented in Figure 16. We notice that the low quality disparity maps contain two types of errors (noise): they either miscalculate the pixel disparity value, or they contain missing values in the form of impulse noise. The stereo tracker performance when employing noisy disparity maps was tested for the videos 2, 3 and 9. The ATA of the stereo tracker with employment of high-quality and noisy disparity maps is shown in Table 4. We notice that, in videos 2 and 3, the noisy disparity maps cause a slight drop in the algorithm performance with respect to high-quality disparity maps. However, the ATA is still better than the ATA of the monocular state of the art appearance-based trackers. On the other hand, the noisy disparity map causes an increase in the ATA in the case of video 9.

The robustness of the stereo tracker performance when using noisy disparity maps is due to the fact that the miscalculated disparity values do not cause significant changes in the ROI mean disparity values and, consequently, the correspondence between the left and right channels remains intact. Moreover, the fast disparity estimation algorithm [57] fails to extract the disparity values at certain pixels, mainly ones that lie on the object contour. This is the case of video 9, as illustrated in Figure 16c. This provides ad-

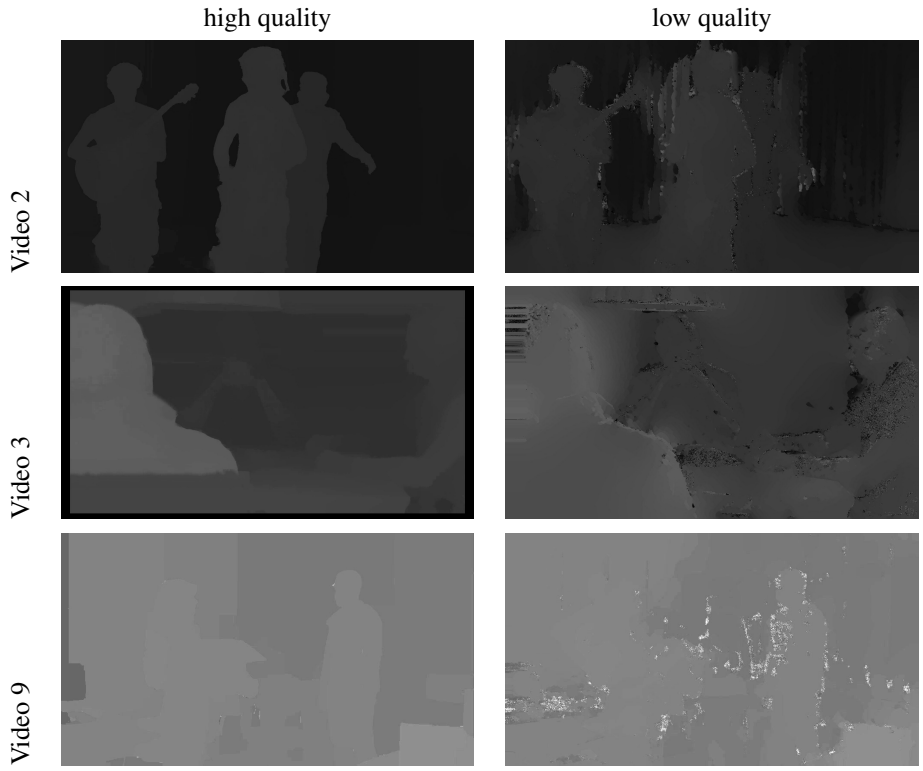


Figure 16: Disparity maps estimated using a high quality and a poor disparity estimation method in videos 2, 3 and 9.

ditional information about the object texture that increases the algorithm performance. However, the algorithm is not immune to extremely noisy disparity maps, since the existence of too much disparity noise deteriorates the ability to discriminate the object from its background through 2D color-disparity histograms. In cases of very noisy disparity maps, the threshold η, defined in the last paragraph of Section 3.2, should be set to zero, i.e., only luminance information should be considered for determining the new object position.

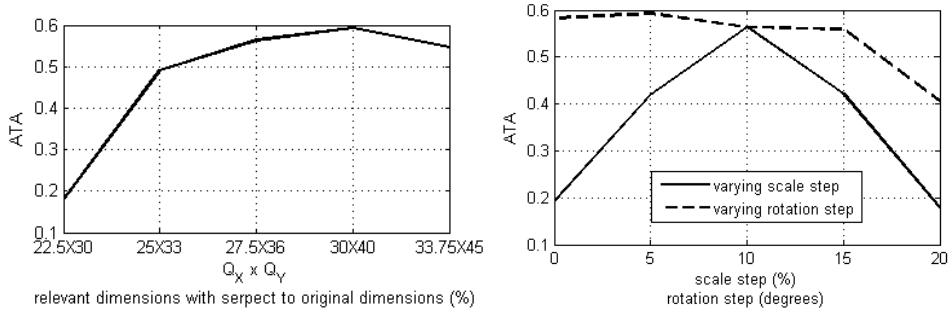
4.6. Parameter setting

We found experimentally that the parameter settings described in Subsection 4.1 allow achieving good tracking results. Therefore, these values are recommended on a wide variety of stereo object tracking scenarios. The only parameters that require

Table 4: ATA of the stereo tracker when high and low quality disparity maps are employed.

ATA	High quality disparity map	Noisy disparity map
video 2	**0.5633**	0.5328
video 3	**0.7136**	0.6649
video 9	0.5993	**0.6322**

tuning are the resized object ROI dimensions $Q_x \times Q_y$, the search region dimensions $R_x \times R_y$, and the angle and scale parameters. The resized object ROI dimensions $Q_x \times Q_y$ are set approximately to be equal to 30%-40% of the original object dimensions in the first video frame. They depend on object texture. If the object texture contains many details, then we try not to down-scale it too much, in order to preserve its salient characteristics. The tracking performance for video 2, for varying values of $Q_x \times Q_y$ is depicted in Figure 17a. We notice that the tracking performance remains relatively constant for a rather large range of $Q_x \times Q_y$, roughly from $25 \times 33\%$ to $33.75 \times 45\%$ of the original object dimensions, respectively and that the performance drops rapidly, when $Q_x \times Q_y$ values drop to $22.5 \times 30\%$ of the original object dimensions, respectively. The search region dimensions $R_x \times R_y$ are proportional to the resized object ROI dimensions. They depend on object velocity. If the object translation is expected to be big, the search region dimensions are set approximately to $R_x \times R_y = 2(Q_x \times Q_y)$. If the object translation is moderate, then a typical value for the search region dimensions is approximately $R_x \times R_y = 1.5(Q_x \times Q_y)$. Finally, the scale and rotation parameters indicate the extent of scale and rotation expected for the object between successive frames. Typical values are $\pm 10\%$ for scale and ± 10 degrees for rotation. If limited size and/or orientation changes are expected, then smaller values for the scale and rotation parameters can be selected, e.g., $\pm 5\%$ for scale and ± 5 degrees for rotation. If size and orientation changes are very fast, then larger values, e.g., $\pm 20\%$ for scale and ± 20 degrees for rotation can be selected. The tracking performance for video 2, for varying values of the scale and rotation parameters is depicted in Figure 17b. We notice that the value of the scale parameter has a larger effect in the tracking accuracy compared to that of the rotation parameter. Good tracking accuracy



(a) (b)

Figure 17: Stereo tracker performance (ATA) for varying parameters of a) the resized object ROI dimensions $Q_x \times Q_y$ and b) for the scale and rotation parameters for video 2.

is obtained for moderate values of the scale parameter (between $5 - 15\%$) and small values of the rotation parameter (between 0-15 degrees). The tracking performance drops significantly when no change in object scale is taken into account (0% scale) or when the scale parameter takes large values, e.g., 20%.

4.7. Computational Complexity

The algorithm runs at 3 fps on high definition stereo videos with frame dimensions 1920×1080 pixels on a computer with a 2.8 GHz processor and 4 GB RAM, without any particular C code optimization. The algorithm speed can be doubled on stereo videos with lower resolution. The extraction of the LSKs and the computation of their similarity takes approximately 40% of the computational time. Since the color-disparity histograms and the LSK features of the candidate object ROIs can be computed in parallel, the tracking speed can be significantly increased if the algorithm is optimized for execution in multiple cores or implemented so as to run in GPUs. Real time performance can be expected in this case.

5. Conclusion

In this paper, a novel method for visual object tracking in stereo videos was proposed. The method employs Local Steering Kernel descriptors and 2-dimensional

color-disparity histograms for the representation of the object appearance. Disparity information is also exploited in order to match corresponding regions in the left and right video frames. The algorithm performs online learning of the object model, i.e., the significant changes in the object appearance, due to scale, rotation, or deformation, are identified and the object model is updated. The object appearance changes are identified simultaneously in the left and right video frames, ensuring the stereo consistency of the resulting bounding boxes.

The proposed framework performs stereo object tracking and it is suitable for application in 3D movies, 3D television programs and 3D content captured from commercial stereo cameras. The only requirement is the availability of disparity information. Disparity estimation may be performed prior to tracking. However, the proposed stereo tracker is an appearance-based tracker, therefore it suffers from the same limitations all appearance-based trackers have, such as, sensitivity to motion blur caused by the object's sudden movement, or inability to track objects that undergo strong deformations.

Experimental results proved the effectiveness of the proposed stereo LSK tracker in tracking objects under geometrical transformations and partial occlusion, as well as in tracking rather slowly deforming/articulated objects. The tracking results of the proposed stereo tracking framework were compared to the tracking results of a monocular tracker which also uses LSK descriptors for texture representation and color-histogram information instead of 2-D color-disparity histogram. Experimental results showed the superiority of the stereo tracking scheme to the monocular one.

Future work is directed towards the exploitation of depth data, captured from low cost off-the-shelf depth cameras, such as the Kinect, as well as, the utilization of feature-based object descriptors instead of the appearance-based LSKs, such as SIFT, SURF and BRIEF in the tracking framework that may enhance the tracking performance. Moreover, other object/background discrimination methods, based on, e.g., stereoscopic optical flow, will be examined. Finally, the extension of the proposed algorithm in object tracking from multiple stereo cameras is considered, by enriching the object model with the object appearances in all stereo cameras and by restricting the object position in the various stereo cameras from the epipolar geometry of the system.

6. Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 2876 74 (3DTVS). This publication reflects only the author's views. The European Union is not liable for any use that may be made of the information contained therein. The authors would like to thank the Fraunhofer Heinrich Hertz Institute for providing the stereo videos 1-6 and the corresponding disparity maps used in the experimental section. Videos 1-5 and video 6 have been produced in EU funded projects MUSCADE and 3D4YOU, respectively.

References

[1] K. Ni, F. Dellaert, Stereo tracking and three-point/one-point algorithms - a robust approach in visual odometry, in: IEEE International Conference on Image Processing, 2006, pp. 2777 –2780. doi:10.1109/ICIP.2006.313123.

[2] A. Johnson, S. Goldberg, Y. Cheng, L. Matthies, Robust and efficient stereo feature tracking for visual odometry, in: IEEE International Conference on Robotics and Automation, 2008, pp. 39 –46. doi:10.1109/ROBOT.2008.4543184.

[3] J. P. Barreto, L. Perdigoto, R. Caseiro, H. Araujo, Active stereo tracking of $n \leq 3$ targets using line scan cameras, IEEE Transactions on Robotics 26 (3) (2010) 442–457. doi:10.1109/TRO.2010.2047300.

[4] S.-W. Shih, J. Liu, A novel approach to 3-d gaze tracking using stereo cameras, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 34 (1) (2004) 234 –245. doi:10.1109/TSMCB.2003.811128.

[5] J. Wang, G. Bebis, R. Miller, Robust video-based surveillance by integrating target detection with tracking, in: IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2006, pp. 137–145. doi:10.1109/CVPRW.2006.180.

[6] C. Yang, R. Duraiswami, L. Davis, Efficient mean-shift tracking via a new similarity measure, in: IEEE Conference on Computer Vision and Pattern Recognition., Vol. 1, 2005, pp. 176 – 183. doi:10.1109/CVPR.2005.139.

[7] Y. Wang, O. Lee, Active mesh-a feature seeking and tracking image sequence representation scheme, IEEE Transactions on Image Processing 3 (1994) 610–624. doi:10.1109/83.334982.

[8] A. Yilmaz, X. Li, M. Shah, Contour-based object tracking with occlusion handling in video acquired using mobile cameras, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1531 –1536. doi:10.1109/TPAMI.2004.96.

[9] L.-Q. Xu, P. Puig, A hybrid blob- and appearance-based framework for multi-object tracking through complex occlusions, in: 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance., 2005, pp. 73 – 80. doi:10.1109/VSPETS.2005.1570900.

[10] P. Sand, S. Teller, Particle video: Long-range motion estimation using point trajectories, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2195 – 2202. doi:10.1109/CVPR.2006.219.

[11] A. Czyzewski, P. Dalka, Examining kalman filters applied to tracking objects in motion, International Workshop on Image Analysis for Multimedia Interactive Services 0 (2008) 175–178. doi:10.1109/WIAMIS.2008.23.

[12] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, IEEE Transactions on Signal Processing 50 (2) (2002) 174 –188. doi:10.1109/78.978374.

[13] C. Stauffer, W. E. L. Grimson, Adaptive background mixture models for real-time tracking, Vol. 2, 1999, pp. 246–252. doi:10.1109/CVPR.1999.784637.

[14] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM Computing Surveys 38 (4). doi:10.1145/1177352.1177355.

[15] H. Yang, L. Shao, F. Zheng, L. Wang, Z. Song, Recent advances and trends in visual tracking: A review, Neurocomputing 74 (18) (2011) 3823–3831. doi:http://dx.doi.org/10.1016/j.neucom.2011.07.024.

[16] H. J. Seo, P. Milanfar, Training-free, generic object detection using locally adaptive regression kernels, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1688–1704. doi:10.1109/TPAMI.2009.153.

[17] O. Zoidi, A. Tefas, I. Pitas, Visual object tracking based on local steering kernels and color histograms, IEEE Transactions on Circuits and Systems for Video Technology 23 (5) (2013) 870–882. doi:10.1109/TCSVT.2012.2226527.

[18] O. Zoidi, N. Nikolaidis, I. Pitas, Appearance based object tracking in stereo sequences, in: 38th International Conference on Acoustics, Speech, and Signal Processing.

[19] L. Cai, L. He, Y. Xu, Y. Zhao, X. Yang, Multi-object detection and tracking by stereo vision, Pattern Recognition 43 (12) (2010) 4028 – 4041. doi:http://dx.doi.org/10.1016/j.patcog.2010.06.012.

[20] T. Zhao, M. Aggarwal, R. Kumar, H. Sawhney, Real-time wide area multi-camera stereo tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 976 – 983. doi:10.1109/CVPR.2005.296.

[21] M. Harville, Stereo person tracking with adaptive plan-view templates of height and occupancy statistics, Image and Vision Computing 22 (2) (2004) 127 – 142.

[22] R. Muñoz Salinas, E. Aguirre, M. García-Silvente, People detection and tracking using stereo vision and color, Image and Vision Computing 25 (6) (2007) 995–1007. doi:http://dx.doi.org/10.1016/j.imavis.2006.07.012.

[23] S. Bahadori, L. Iocchi, G. R. Leone, D. Nardi, L. Scozzafava, Real-time people localization and tracking through fixed stereo vision 26 (2). doi:10.1007/11504894_6.

[24] F. Tang, M. Harville, H. Tao, I. Robinson, Fusion of local appearance with stereo depth for object tracking, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2008, pp. 1 –8. doi:10.1109/CVPRW.2008.4563036.

[25] C. G. Keller, M. Enzweiler, M. Rohrbach, D. F. Llorca, C. Schnorr, D. M. Gavrila, The benefits of dense stereo for pedestrian detection, IEEE Transactions on Intelligent Transportation Systems 12 (4) (2011) 1096–1106. doi:10.1109/TITS.2011.2143410.

[26] C. Li, L. Lu, G. D. Hager, J. Tang, H. Wang, Robust object tracking in crowd dynamic scenes using explicit stereo depth, in: Computer Vision–ACCV 2012, Springer, 2013, pp. 71–85. doi:10.1007/978-3-642-37431-9_6.

[27] G. Catalin, S. Nedevschi, Object tracking from stereo sequences using particle filter, in: 4th International Conference on Intelligent Computer Communication and Processing, 2008, pp. 279–282. doi:10.1109/ICCP.2008.4648386.

[28] L. Zhao, C. E. Thorpe, Stereo-and neural network-based pedestrian detection, IEEE Transactions on Intelligent Transportation Systems 1 (3) (2000) 148–154. doi:10.1109/6979.892151.

[29] D. M. Gavrila, S. Munder, Multi-cue pedestrian detection and tracking from a moving vehicle, International journal of computer vision 73 (1) (2007) 41–59. doi:10.1007/s11263-006-9038-7.

[30] A. Gaschler, D. Burschka, G. Hager, Epipolar-based stereo tracking without explicit 3d reconstruction, in: 20th International Conference on Pattern Recognition, 2010, pp. 1755 –1758. doi:10.1109/ICPR.2010.434.

[31] R. Muñoz Salinas, M. García-Silvente, R. Medina Carnicer, Adaptive multi-modal stereo people tracking without background modelling, Journal of Vision Comunication and Image Representation 19 (2) (2008) 75–91. doi:http://dx.doi.org/10.1016/j.jvcir.2007.07.004.

[32] T. Darrell, G. Gordon, M. Harville, J. Woodfill, Integrated person tracking using stereo, color, and pattern detection, IEEE conference on Computer Vision and Patern Recognition (1998) 601–608doi:10.1109/CVPR.1998.698667.

[33] K.-H. Bae, J.-S. Koo, E.-S. Kim, A new stereo object tracking system using disparity motion vector, Optics Communications 221 (1-3) (2003) 23 – 35. doi:http://dx.doi.org/10.1016/S0030-4018(03)01453-6.

[34] E. Parrilla, J. Riera, J.-R. Torregrosa, J.-L. Hueso, Handling occlusion in object tracking in stereoscopic video sequences, Mathematical and Computer Modelling 50 (56) (2009) 823 – 830. doi:http://dx.doi.org/10.1016/j.mcm.2008.12.021.

[35] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, D. Cremers, Stereoscopic scene flow computation for 3d motion understanding, International Journal of Computer Vision 95 (1) (2011) 29–51. doi:10.1007/s11263-010-0404-0.

[36] T. Nakamura, Real-time 3-d object tracking using kinect sensor, in: IEEE International Conference on Robotics and Biomimetics (ROBIO), 2011, pp. 784–788. doi:10.1109/ROBIO.2011.6181382.

[37] I. Oikonomidis, N. Kyriazis, A. A. Argyros, Efficient model-based 3d tracking of hand articulations using kinect, in: Proceedings of the British Machine Vision Conference, 2011, pp. 101.1–101.11. doi:http://dx.doi.org/10.5244/C.25.101.

[38] Y. Park, V. Lepetit, W. Woo, Texture-less object tracking with online training using an rgb-d camera, in: 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2011, pp. 121–126. doi:10.1109/ISMAR.2011.6092377.

[39] L. Xia, C.-C. Chen, J. Aggarwal, Human detection using depth information by kinect, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2011, pp. 15–22. doi:10.1109/CVPRW.2011.5981811.

[40] L.-P. Morency, T. Darrell, Stereo tracking using icp and normal flow constraint, in: 16th International Conference on Pattern Recognition, Vol. 4, 2002, pp. 367 – 372. doi:10.1109/ICPR.2002.1047472.

[41] P. Besl, H. McKay, A method for registration of 3-d shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239 –256. doi:10.1109/34.121791.

[42] Y. Chen, G. Medioni, Object modeling by registration of multiple range images, in: IEEE International Conference on Robotics and Automation, Vol. 3, 1991, pp. 2724 –2729. doi:10.1109/ROBOT.1991.132043.

[43] E. Trucco, A. Verri, Introductory Techniques for 3-D Computer Vision, Vol. 93, Prentice Hall, 1998.

[44] Z.-F. Wang, Z.-G. Zheng, A region based stereo matching algorithm using cooperative optimization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1 –8. doi:10.1109/CVPR.2008.4587456.

[45] Q. Yang, L. Wang, R. Yang, H. Stewenius, D. Nister, Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (3) (2009) 492 –504. doi:10.1109/TPAMI.2008.99.

[46] A. Klaus, M. Sormann, K. Karner, Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure, in: 18th International Conference on Pattern Recognition, Vol. 3, 2006, pp. 15 –18. doi:10.1109/ICPR.2006.1033.

[47] A. Hosni, M. Bleyer, M. Gelautz, C. Rhemann, Local stereo matching using geodesic support weights, in: 16th IEEE International Conference on Image Processing, 2009, pp. 2093 –2096. doi:10.1109/ICIP.2009.5414478.

[48] G. Welch, G. Bishop, An introduction to the kalman filter, in: University of North Carolina at Chapel Hill, Tech. Rep. TR95041, 2000.

[49] N. Atzpadin, P. Kauff, O. Schreer, Stereo analysis by hybrid recursive matching for real-time immersive video conferencing, IEEE Transactions on Circuits and Systems for Video Technology, 14 (3) (2004) 321 – 334. doi:10.1109/TCSVT.2004.823391.

[50] C. Riechert, F. Zilly, P. Kauff, "real time depth estimation using line recursive matching, in: European Conference on Visual Media Production, 2011.

[51] V. Kolmogorov, R. Zabih, Computing visual correspondence with occlusions using graph cuts, in: Proceedings of IEEE Conference of Computer Vision, Vol. 1, 2001, pp. 508–515. doi:10.1109/ICCV.2001.937668.

[52] S. Zhou, R. Chellappa, B. Moghaddam, Visual tracking and recognition using appearance-adaptive models in particle filters, IEEE Transactions on Image Processing 13 (2004) 1434–1456. doi:10.1109/TIP.2004.836152.

[53] X. Mei, H. Ling, Robust visual tracking using l_1 minimization, in: IEEE 12th International Conference on Computer Vision, 2009, pp. 1436 –1443. doi:10.1109/ICCV.2009.5459292.

[54] B. Babenko, M.-H. Yang, S. Belongie, Visual tracking with online multiple instance learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 983 –990. doi:10.1109/CVPR.2009.5206737.

[55] K. Zhang, L. Zhang, M.-H. Yang, Real-time compresive tracking, in: European Conference on Computer Vision, 2012, pp. 864–877. doi:10.1007/978-3-642-33712-3_62.

[56] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, J. Zhang, Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 319–336. doi:10.1109/TPAMI.2008.57.

[57] S. Kosov, T. Thormhlen, H.-P. Seidel, Accurate real-time disparity estimation with variational methods, in: Advances in Visual Computing, Vol. 5875 of Lec-

ture Notes in Computer Science, 2009, pp. 796–807. doi:10.1007/978-3-642-10331-5_74.